What is claimed is:

1. A method for use with a database system that stores a join view associated with plural base relations, comprising:

receiving modification operations that modify at least two of the base relations of the join view; and

re-ordering the received modification operations to avoid execution of modification operations of more than one of the at least two base relations at one time in the database system.

2. The method of claim 1, wherein receiving the modification operations comprises receiving a first modification operation to modify a first base relation of the join view, and a second modification operation to modify a second base relation of the join view,

wherein re-ordering the modification operations comprises:

determining that the first modification operation conflicts with the second modification operation based on the first and second modification operations modifying more than one base relation of the join view; and

selecting one of the first and second modification operations for execution in the database system.

3. The method of claim 1, wherein selecting one of the first and second modification operations comprises selecting the first modification operation,

the method further comprising storing the second modification operation in a queue.

4. The method of claim 3, further comprising waiting for the first modification operation to complete execution before scheduling the second modification operation for operation.

5. The method of claim 4, further comprising receiving a third modification operation to modify the first base relation of the join view, the method further comprising:

storing the third modification operation in the queue; and

scheduling the third modification operation for execution in the database system ahead of the second modification operation.

6. The method of claim 2, further comprising:

submitting modification operations to the database system in plural sessions;

receiving a third modification operation that modifies the first base relation;

determining that the first modification operation modifies a first portion of the first base relation, and the third modification operation modifies a second, distinct portion of the first base relation; and

in response to determining that the first and third modification operations modify distinct portions of the first base relation, submitting the first and third modification operations through different sessions for concurrent execution in the database system.

7. The method of claim 2, further comprising:

receiving additional modification operations on the first base relation; and

grouping the modification operations on the first base relation into a transaction; and

submitting the transaction to the database system for execution.

8. The method of claim 1, further comprising:

in response to a modification operation to modify one of the base relations, placing an exclusive lock on the one base relation, and placing a predefined lock on the join view,

the predefined lock conflicting with either a shared lock or an exclusive lock placed on the join view but not conflicting with another predefined lock placed on the join view.

9. The method of claim 1, further comprising:

storing pending modification operations in plural queues corresponding to respective plural sessions of the database system; and

selecting one of the pending modification operations from the queues to schedule for execution in the database system based on whether the one pending modification operation conflicts with one or more executing modification operations in the database system.

10. The method of claim 9, further comprising determining that the one pending modification operation conflicts with the one or more executing modification operations in response to determining that the one pending modification operation modifies a different one of the base relations of the join view than a base relation of the join view modified by an executing modification operation.

11. The method of claim 9, further comprising applying a technique to prevent starvation of one of the pending modification operations in response to determining that the one pending modification operation has been in one of the queues for longer than a predetermined time period.

12. An article comprising at least one storage medium containing instructions that when executed cause a system to:

receive modification operations that modify at least two of the base relations of a join view; and

re-order the received modification operations to avoid concurrent execution of modification operations of more than one of the at least two base relations of the join view.

13. The article of claim 12, wherein receiving the modification operations comprises receiving a first modification operation to modify a first base relation of the join view, and a second modification operation to modify a second base relation of the join view,

wherein re-ordering the modification operations comprises:

determining that the first modification operation conflicts with the second modification operation based on the first and second modification operations modifying more than one base relation of the join view; and

selecting one of the first and second modification operations for execution in the database system.

14. The article of claim 13, wherein selecting one of the first and second modification operations comprises selecting the first modification operation,

the instructions when executed causing the system to further store the second modification operation in a queue.

15. The article of claim 14, wherein the instructions when executed cause the system to wait for the first modification operation to complete execution before scheduling the second modification operation for operation.

16. The article of claim 13, wherein the instructions when executed cause the system to:

submit modification operations to a database system in plural sessions;

receive a third modification operation that modifies the first base relation;

determine that the first modification operation modifies a first portion of the first base relation, and the third modification operation modifies a second, distinct portion of the first base relation; and

in response to determining that the first and third modification operations modify distinct portions of the first base relation, submit the first and third modification operations through different sessions for concurrent execution in the database system.

17. The article of claim 13, wherein the instructions when executed cause the system to:

receive additional modification operations on the first base relation; and

group the modification operations on the first base relation into a transaction; and

submit the transaction to a database system for execution.

18. The article of claim 12, wherein the instructions when executed cause the system to:

in response to a modification operation to modify one of the base relations, place an exclusive lock on the one base relation, and place a predefined lock on the join view,

the predefined lock conflicting with either a shared lock or an exclusive lock placed on the join view but not conflicting with another predefined lock placed on the join view.

19. A system comprising:

a controller to:

receive modification operations to modify plural base relations of a join view, the modification operations comprising modification operations to modify a first base relation of the join view, and modification operations to modify a second base relation of the join view; and

re-order the received modification operations to avoid concurrent execution of modification operations of more than one of the plural base relations of the join view,

the re-ordering to cause modification operations on the first base relation of the join view to be scheduled for execution, and to cause modification operations on the second base relation to be queued for execution after completion of the modification operations on the first base relation.

20. The system of claim 19, wherein the controller is adapted to identify the modification operations on the second base relation as conflicting with the modification operations on the first base relation in response to determining that the modification operations on the second base relation are modifying a different base relation of the join view than the modification operations on the first base relation.

21. The system of claim 20, wherein the system comprises a first system, and wherein the controller is adapted to open plural sessions with a database system separate from the first system,

the controller to further:

determine that the modification operations on the first base relation modify distinct portions of the first base relation; and

in response to determining that the modification operations on the first base relation modify distinct portions of the first base relation, submit the modification operations on the first base relation through different sessions for concurrent execution in the database system.

22. The system of claim 20, wherein the system comprises a first system, wherein the modification operations on the first base relation comprises modification operations of a first tuple of the first base relation, and wherein the controller is adapted to:

group the modification operations on the first tuple of the first base relation into a transaction; and

submit the transaction to a database system separate from the first system for execution.

23. The system of claim 19, wherein the controller comprises a load utility to submit the modification operations to a database system.

24. The system of claim 23, wherein the load utility comprises a continuous load utility.

25. The system of claim 23, wherein the load utility comprises a first load utility, and the controller comprises a second load utility to concurrently submit other modification operations to the database system.

26. The system of claim 25, further comprising plural platforms on which corresponding first and second load utilities are executable.